# Combining Large-Scale Unlabeled Corpus and Lexicon for Chinese Polysemous Word Similarity Computation

Huiwei Zhou(✉), Chen Jia, Yunlong Yang, Shixian Ning,
Yingyu Lin, and Degen Huang

School of Computer Science and Technology, Dalian University of Technology,
Dalian 116024, Liaoning, China
{zhouhuiwei,huangdg}@dlut.edu.cn,
{jiachen,SDyyl_1949,ningshixian}@mail.dlut.edu.cn,
lyydut@sina.com

**Abstract.** Word embeddings have achieved an outstanding performance in word similarity measurement. However, most prior works focus on building models with one embedding per word, neglect the fact that a word can have multiple senses. This paper proposes two sense embedding learning methods based on large-scale unlabeled corpus and Lexicon respectively for Chinese polysemous words. The corpus-based method labels the senses of polysemous words by clustering the contexts with *tf-idf* weight, and using the HowNet to initialize the number of senses instead of simply inducing a fixed number for each polysemous word. The lexicon-based method extends the *AutoExtend* to Tongyici Cilin with some related lexicon constraints for sense embedding learning. Furthermore, these two methods are combined for Chinese polysemous word similarity computation. The experiments on the Chinese Polysemous Word Similarity Dataset show the effectiveness and complementarity of our two sense embedding learning methods. The final Spearman rank correlation coefficient achieves 0.582, which outperforms the state-of-the-art performance on the evaluation dataset.

**Keywords:** Sense embeddings · Chinese word similarity evaluation · Chinese polysemous words · Large-scale unlabeled corpus · Lexicon

## 1 Introduction

Many Nature Language Processing tasks benefit from word embeddings [1] because they help capture syntactic and semantic characteristics of words by projecting words into a low-dimensional vector space. However, most of the previous researches in word embeddings associate each word with a single embedding. Such single embedding representation method is forced to express polysemous word with an uneasy central vector between its various meanings [2]. This will lead to a text understanding problem, since polysemous words may have different meanings in different contexts.

In recent years, there has been an increasing interest in learning sense embeddings for polysemous words from large-scale unlabeled corpus [3, 4]. These researches learn

J. Wen et al. (Eds.): CCIR 2017, LNCS 10390, pp. 198–210, 2017.
https://doi.org/10.1007/978-3-319-68699-8_16

sense embeddings based on two fundamental steps: (1) Label the sense of the polysemous words by clustering the contexts. (2) Learn sense embeddings for polysemous words. Such methods simply include uniform *K* senses for all polysemous words, ignoring the real sense number of each polysemous word. Besides, the simple context representations without regard to the word frequency limit the performance of clusters, and therefore prevent improving sense embeddings. Meanwhile, sense embeddings trained on large-scale unlabeled corpus suffers from the unsatisfied quality of embeddings of low frequency senses.

This paper proposes a sense embedding learning method based on large-scale unlabeled corpus for Chinese polysemous words. The proposed corpus-based method clusters the contexts of polysemous words based with *tf-idf* (Term Frequency-Inverse Document Frequency) weight [5], and using the HowNet [6] to initialize the number of senses for each polysemous word. Then, to address low frequency senses, we extend the *AutoExtend* [7] to the Tongyici Cilin [8] with some related lexicon constraints and use it as an annotated knowledge base to learn lexicon-based sense embeddings. Finally, these two methods are combined for Chinese polysemous word similarity computation [9]. The experimental results show that the corpus-based and lexicon-based methods are both effective for capturing the sense-level word similarities, and their combination could further improve similarity computation performance.

## 2 Related Work

Bengio et al. [10] first propose to learn word embeddings by using neural networks. Continuous Word2vec model [11, 12] and GloVe model [13] are the two efficient models for learning word embeddings. However, representing each word with a single embedding fails to capture polysemy. Reisinger et al. [3] and Huang et al. [4] introduce a kind of sense embedding learning method by clustering the contexts of polysemous words and then learning embeddings of the sense-labeled words. Neelakantan et al. [14] improve sense embeddings by performing word sense clustering and embedding learning jointly. But these methods usually assume that contexts are directly represented as sum (or average) of the surrounding words' vectors, neglecting that the context words do not contribute equally to the sense of the polysemous word. Zheng et al. [15] tackle this issue by learning the context representations with a neural network architecture before learning sense embeddings. Guo et al. [9] propose a novel approach for Chinese sense embedding leaning by exploiting bilingual parallel data. However, these corpus-based models are generally unable to learn some low frequency senses which rarely occurrence in regular corpus and bilingual parallel data.

Lexicon-based models drawing this issue by using sense-specific knowledge. Chen et al. [16] use WordNet glosses to learn sentence-level embeddings as the initializations of sense embeddings. Rothe et al. [7] take advantage of the architecture of WordNet to extend word embeddings to embeddings of synsets and lexemes. However, these models generally suffer for inaccurate semantic representations for sense items in the lexicon. Pei et al. [17] combine semantic lexicon and word embeddings to compute Chinese word similarity, but they neglect polysemy.

As discussed above, there are only a few researches on learning sense embeddings, and almost all of them focus on English words. We address these issues and introduce a corpus-based and a lexicon-based sense embedding learning methods, which are demonstrated effective and complementary for Chinese polysemous word similarity computation.

# 3 Methods

Figure 1 briefly illustrates the general architecture of our Chinese polysemous word similarity computation system. It consists of three components: (1) the corpus-based sense embedding learning, (2) the lexicon-based sense embedding learning and (3) the similarity computation and combination.

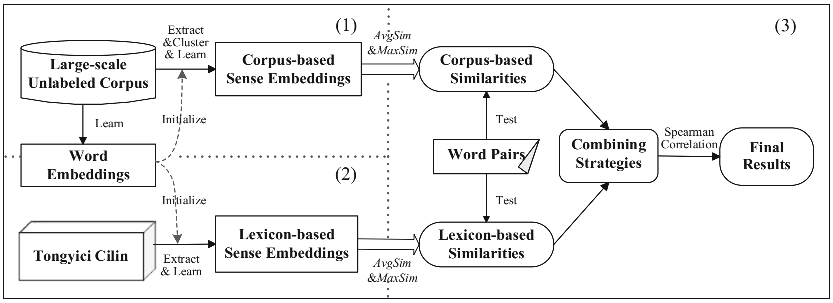

**Fig. 1.** Architecture of our Chinese polysemous word similarity computation system.

## 3.1 Corpus-Based Sense Embedding Learning

Based on an enlightening idea that word sense is associated with its context [3, 4], the corpus-based sense embedding learning uses the contexts of the polysemous words to distinguish senses. The detail of our method can be described into three sections:

- Context Representation Extraction: Context key words of the current polysemous word are extracted based on *tf-idf* weight for context representations.
- Sense Clustering: The context representations are clustered according to the sense number of each polysemous word to generate clusters, which are then used to label the sense of polysemous words.
- Sense Embedding Learning: Regarding each sense of a word as a new word, the sense embeddings of polysemous words and the single embeddings of the other monosemous words are learned and unified in one vector space with the GloVe model [13].

**Context Representation Extraction.** Consider a word $w_t$ and its contexts $c_t = \{w_{t-R_t}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+R_t}\}$, where $R_t$ represents the semi-width of the context window. The context representation of word $w_t$ is the sum of its context word embeddings:

$$v_{context}(w_t) = \sum_{w \in c_t} v_g(w) \tag{1}$$

where $v_g(w)$ is the pre-trained word embedding of $w \in c_t$.

Although the large context window contains the comprehensive meaning of the contexts, too many context words also obfuscate the representation of the meaning. For example, some non-essential words such as "是 (is or are)" and "有 (has or have)" definitely have a negative effect on the context representations.

This paper defines a selectivity factor $R_t^*$ to choose the key words in the contexts. To be specific, we use *tf-idf* to compute the weights of context words [5]. And then choose $R_t^*(R_t^* \leq R_t)$ words $c_t^* = \{w_{t-R_t^*}^*, \ldots, w_{t-1}^*, w_{t+1}^*, \ldots, w_{t+R_t^*}^*\}$ with the largest $R_t^*-th$ weights as the context representations:

$$v_{context}(w_t) = \sum_{w \in c_t^*} v_g(w) \tag{2}$$

**Sense Clustering.** For each polysemous word, the extracted context representations are then used for sense clustering. This paper uses *k*-means clustering algorithm [3].

Here, assume that $S_j$ $(j = 1, 2, \ldots, K)$ is the $j^{th}$ cluster whose center is $\mu_j$. $K$ is the number of senses. Instead of fixing the number of all clusters to a single parameter $K$, we use the HowNet [6] to initialize the number of senses for each polysemous word. The clustering rule can be represented as:

$$J = \sum_{j=1}^{K} \sum_{v_{context} \in S_j} sim(\mu_j, v_{context}) \tag{3}$$

where $sim(\cdot, \cdot)$ is cosine similarity. The object of clustering is to maximize the clustering rule $J$. And then each polysemous word in the unlabeled corpus is labeled a sense tag according to the sense clusters.

**Sense Embedding Learning.** Once a sense labeled corpus are generated, the GloVe model [13] is used to learn the sense embeddings of the polysemous words and the single embeddings of the other monosemous words in one vector space. The GloVe model is directly based on the statistics of word (or sense) occurrences in the sense labeled corpus. Let the matrix of word-sense co-occurrence counts be denoted by $X$, whose entries $X_{ij}$ tabulate the number of times that word (or sense) $j$ occurs in the context of word (or sense) $i$. And $X_i = \sum_k X_{ik}$ is the number of times that any word (or sense) appears in the context of word (or sense) $i$. Finally, $P_{ij} = P(j|i) = X_{ij}/X_i$ is the probability that word (or sense) $j$ appears in the context of word (or sense) $i$.

We use a simple example to show how certain senses of a polysemous word can be extracted directly from co-occurrence probabilities with various probe words $k$. Consider two senses $i$ = 打 (*fetch*) and $j$ = 打 (*hit*) of the word "打". For words $k$ related to $i$ = 打 (*fetch*) but not $j$ = 打 (*hit*), say $k$ = 水 (water), we expect the ratio $P_{ik}/P_{jk}$ will be large. However, for words $k$ related to $j$ = 打 (*hit*) but not $i$ = 打 (*fetch*), say $k$ = 人 (man), the ratio should be small.

### 3.2 Lexicon-Based Sense Embedding Learning

We utilize the HIT-CIR Tongyici Cilin (Extended) [8] as the lexical resource to learn sense embeddings of Chinese polysemous words. Figure 2 briefly illustrates the process of lexicon-based sense embedding learning.

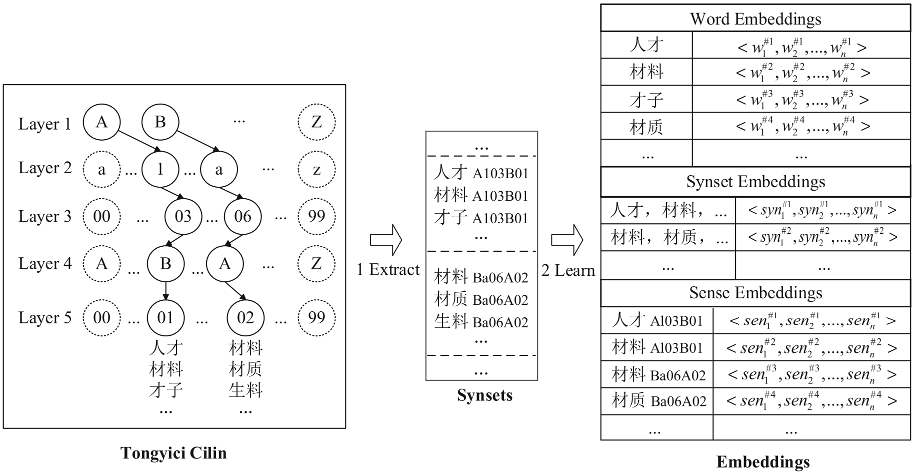
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
A
B
...
Z
a
1
a
z
00
03
06
99
A
B
A
Z
00
01
02
99
人才
材料
才子
...
材料
材质
生料
...
Tongyici Cilin
1 Extract
...
人才 A103B01
材料 A103B01
才子 A103B01
...
材料 Ba06A02
材质 Ba06A02
生料 Ba06A02
...
...
Synsets
2 Learn
Word Embeddings
人才 $< w_1^{\#1}, w_2^{\#1}, ..., w_n^{\#1} >$
材料 $< w_1^{\#2}, w_2^{\#2}, ..., w_n^{\#2} >$
才子 $< w_1^{\#3}, w_2^{\#3}, ..., w_n^{\#3} >$
材质 $< w_1^{\#4}, w_2^{\#4}, ..., w_n^{\#4} >$
...
Synset Embeddings
人才，材料，... $< syn_1^{\#1}, syn_2^{\#1}, ..., syn_n^{\#1} >$
材料，材质，... $< syn_1^{\#2}, syn_2^{\#2}, ..., syn_n^{\#2} >$
...
Sense Embeddings
人才 A103B01 $< sen_1^{\#1}, sen_2^{\#1}, ..., sen_n^{\#1} >$
材料 A103B01 $< sen_1^{\#2}, sen_2^{\#2}, ..., sen_n^{\#2} >$
材料 Ba06A02 $< sen_1^{\#3}, sen_2^{\#3}, ..., sen_n^{\#3} >$
材质 Ba06A02 $< sen_1^{\#4}, sen_2^{\#4}, ..., sen_n^{\#4} >$
...
Embeddings

**Fig. 2.** Lexicon-based sense embedding learning.

The architecture of the Tongyici Cilin is shown in the left part of Fig. 2. Cilin encodes a group of words which have the relationships like "synonym" or "relevance" in a 5-layer hierarchical tree structure. The nodes of the upper 4 layers in the Cilin represent the abstract categories, while only the bottom leaf nodes are the words. A set of synonyms are organized as a synset and coded as the same leaf node in the hierarchical tree. While the polysemous words are coded as the different leaf nodes according to their different senses, that is to say, the same word may have different codes in Cilin. As shown in Fig. 2, the polysemous word "材料" is coded in different synsets "Al03B01 = 人才 (talents), 材料 (stuff) …" and "Ba06A02 = 材料 (material), 材质 (material) …" etc.

We take advantage of the architecture of Cilin by assuming that words with the same code form the synsets, and polysemous words with different synset codes could provide the senses for sense embedding learning. Inspired by *AutoExtend* [7], our model learn sense embedding $sen^{(i,j)} \in \mathbb{R}^n$ of a word together with its word embedding $w^{(i)} \in \mathbb{R}^n$ and the relevant synset embedding $syn^{(j)} \in \mathbb{R}^n$ by premising that $w^{(i)} = \sum_j sen^{(i,j)}$ and $syn^{(j)} = \sum_i sen^{(i,j)}$.

Let $W \in \mathbb{R}^{|W| \times n}$ be a matrix of word embeddings with the size of $|W|$ and $S \in \mathbb{R}^{|S| \times n}$ be a matrix of synset embeddings with the size of $|S|$. We can use a rank 4 tensor $E \in \mathbb{R}^{|S| \times n \times |W| \times n}$ to encode the matrix $W$ into the matrix $S$ and similarly use a rank 4 tensor $D \in \mathbb{R}^{|W| \times n \times |S| \times n}$ to decode the matrix $S$ into the matrix $\overline{W}$. We can state the learning objective under the synset constraints as follows:

$$loss_{syn} = \underset{E,D}{\arg\min} \|D \otimes E \otimes W - W\| \tag{4}$$

where $\otimes$ is tensor product.

The sense embeddings are defined when transitioning from $W$ to $S$ or transitioning from $S$ to $\overline{W}$. Aligning these two representations, we could get the sense constraints as follows:

$$loss_{sen} = \underset{E,D}{\arg\min} \|E \otimes W - D \otimes S\| \tag{5}$$

The above two constraints are very similar to *AutoExtend* architecture [7]. However, the architecture of Cilin leads to a problem that there is a large number of synsets which have only one word (sense). This makes it hard to learn high quality embeddings for single-word synsets. To remedy this problem, we use the relation constraints that the synsets in Cilin which are with the similar codes (either the upper 3 layer codes or 4 layer codes are the same) should have the similar senses and the distribution in the vector space should be close as well. Let $R \in \mathbb{R}^{r \times |S|}$ be the relation matrix, where $r$ is the number of relation tuples. For each row, the dimension corresponding to the original and related synsets are set to 1 and $-1$, respectively. The relation constraints could be written as follows:

$$loss_{rel} = \underset{E}{\arg\min} \|RE \otimes W\| \tag{6}$$

The final training objective is minimizing the sum of synset constraints, sense constraints and relation constraints and more explicitly by:

$$loss = \alpha \cdot loss_{syn} + \beta \cdot loss_{sen} + (1 - \alpha - \beta) \cdot loss_{rel} \tag{7}$$

where $\alpha$ is the weight of synset constraints and $\beta$ is the weight of sense constraints. By giving the three constraints different weights, we can easily learn the sense embeddings and synset embeddings.

### 3.3 Similarity Computation and Combination

**Similarity Computation.** Word similarity is calculated using the *AvgSim* and *MaxSim* metrics [3] respectively:

$$AvgSim(u,v) = \frac{1}{K_u \times K_v} \sum_{i=1}^{K_u} \sum_{j=1}^{K_v} \cos(u^i, v^j) \tag{8}$$

$$MaxSim(u,v) = \max_{1 \le i \le K_u, 1 \le j \le K_v} \cos(u^i, v^j) \tag{9}$$

where $K_u$ and $K_v$ are the number of senses for words $u$ and $v$, $\cos(u^i, v^j)$ is the cosine similarity between the $i^{th}$ sense embedding of word $u$ and the $j^{th}$ sense embedding of word $v$.

**Similarity Combination.** The best similarity results of corpus-based and lexicon-based methods are combined using 5 strategies based on fundamental math operations. For each similarity score pair $Sim_c$ and $Sim_l$ of the corpus-based and lexicon-based model, we calculate a combination score $Sim$ according to the combining strategy. The 5 strategies are defined as follows:

Max

$$Sim = max(Sim_c, Sim_l) \tag{10}$$

Average

$$Sim = \frac{Sim_c + Sim_l}{2} \tag{11}$$

Max and Average

$$Sim = \begin{cases} max(Sim_c, Sim_l) & Sim_c \neq 0,\ Sim_l \neq 0 \\ \frac{Sim_c + Sim_l}{2} & Sim_c = 0 \ or \ Sim_l = 0 \end{cases} \tag{12}$$

Replace 0

$$Sim = \begin{cases} \frac{Sim_c + Sim_l}{2} & Sim_c \neq 0,\ Sim_l \neq 0 \\ Sim_l & Sim_c = 0 \\ Sim_c & Sim_l = 0 \end{cases} \tag{13}$$

Improved Geometric Mean

$$Sim = \begin{cases} \sqrt{Sim_c * Sim_l} & Sim_c \neq 0,\ Sim_l \neq 0 \\ Sim_l & Sim_c = 0 \\ Sim_c & Sim_l = 0 \end{cases} \tag{14}$$

# 4 Experiments

## 4.1 Experimental Setup

We pre-train word embeddings and learn sense embeddings both on two corpora: the Sogou Chinese news corpus[1] and Chinese Wikipedia[2]. All corpora are word segmented by Stanford Word Segmenter[3]. In corpus preprocessing, we remove the common punctuations and some rare symbols. All the numbers in the corpus are replaced by "XXXX". The embedding training datasets is relatively large, with a total of 400 million tokens. With GloVe tool[4], we build a vocabulary of the 150,000 most frequent words whose word frequency are no less than 50. The initial embedding dimension is set to 300, and the sampling window size is set to 10 for the co-occurrence statistics.

In sense embedding learning, the semi-width $R_t$ of the context window is set to 5 and the initialization weights of synset constraints $\alpha$, sense constraints $\beta$ and relation constraints $(1 - \alpha - \beta)$ are set to 0.2, 0.5 and 0.3, respectively.

Following Guo et al. [9] we use Spearman correlation $\rho$ for evaluation. The proposed methods are evaluated on Chinese Polysemous Word Similarity Dataset [9]. Chinese polysemous words in the dataset are extracted according to their sense definitions in HowNet [6]. Then several other polysemous words (from related to unrelated) are selected to form word pairs with each polysemous word. Finally, 401 word pairs are manually annotated with their similarity scores from 0.0 to 10.0 by human judgments. E.g., word "制服 (control)" paired with "征服 (conquer)" gets the similarity score of 8.60. But if paired with "重点 (focus)", the similarity score will only be 0.12.

## 4.2 Evaluation Results

We first evaluate similarity performance by computing the nearest neighbors across corpus-based and lexicon-based sense embeddings respectively. Table 1 lists the nearest word (W/), sense (Sen/) and synset (Syn/) of some senses of polysemous words.

As can be seen from Table 1, for word "开阔", both of the two models could find concrete and abstract senses well. As for word "制服", which has different Part-of-Speech senses, the corpus-based model could distinguish between none and verb senses but the lexicon-based model could not. As for word "材料", which has different aspects of senses, the lexicon-based model could find the sense refers to stuff but the corpus-based could not. This inspires us to combine the two models using some combination strategies.

**The Results of Corpus-Based Sense Embedding Learning.** Figure 3 shows the similarity results, in which the selectivity factor $R_t^*$ is varied from 1 to 5 with an interval of 1. From the Fig. 3 we can see that:

[1] http://www.sogou.com/labs/dl/c.html.

[2] https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2.

[3] http://nlp.stanford.edu/software/segmenter.shtml.

[4] https://nlp.stanford.edu/projects/glove/.

**Table 1.** Nearest neighbors of the senses of some polysemous words.

| Methods | Center word | Nearest neighbors |
|---|---|---|
| Corpus-based | 制服1 (uniform) | W/队服 (jersey), W/礼服 (full dress), W/军装 (military uniform) |
| | 制服2 (control) | W/歹徒 (gangster), W/擒获 (catch), W/抓捕 (arrest) |
| | 材料1 (material) | W/超硬 (super-hard), W/核级 (nuclear), W/金属 (metal) |
| | 材料2 (data) | W/文件 (file), W/资料 (data), W/提交 (submit) |
| | 开阔1 (wide) | W/绿树 (green trees), W/山峦 (mountains), W/环绕 (surround) |
| | 开阔2 (broad) | W/视野 (view), W/实践 (practice), W/学习 (study) |
| Lexicon-based | 制服1 (control) | Sen/歹徒 (gangster), W/歹徒 (gangster), Syn/穿,穿着 (wear) |
| | 材料1 (material) | W/材料 (material), Syn/材料,材质 (material), Sen/材质 (texture) |
| | 材料2 (data) | Syn/资料,材料 (material), W/材料 (material), Sen/资料 (data) |
| | 材料3 (stuff) | W/材料 (material), Sen/人才 (talent), Syn/人才,材料 (talent) |
| | 开阔1 (wide) | W/开阔 (open), Syn/广阔,宽阔 (vast), Sen/广阔 (vast) |
| | 开阔2 (broad) | W/开阔 (wide), Sen/视野 (view), W/视野 (view) |

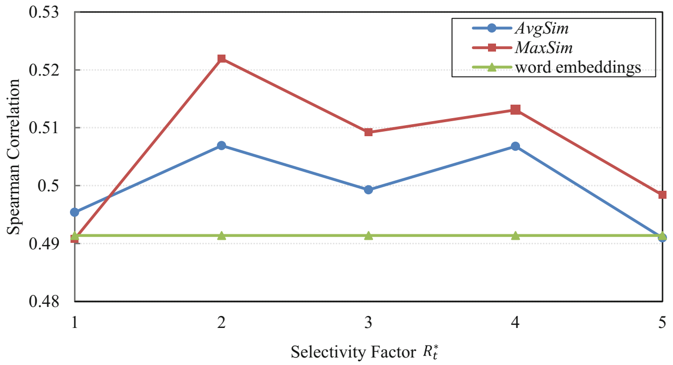

**Fig. 3.** Spearman correlation $\rho$ based on large-scale corpora.

- The sense embeddings learned on the large-scale corpus could improve the performance of the single word embeddings, which only achieves the Spearman correlation $\rho$ of 0.491.
- The best performances are under the conditions $R_t^* = 2$. It indicates that context represented by the words with the largest $R_t^*$-*th tf-idf* weights could improve the cluster performance and therefore obtain the superior performance of sense embeddings than represented by the total words ($R_t^* = 5$). The maximum similarity scores under the conditions $R_t^* = 2$ are used as the corpus-based similarity scores for the combination of the two methods in the following experiments.

**The results of lexicon-based sense embedding learning.** In this section, we use the initialization weights of synset constraints, sense constraints and relation constraints ($\alpha = 0.2$ and $\beta = 0.5$). If the test pairs contain *out of vocabulary* (OOV) words, we simple assign the similarity with the minimum score of 0. *AvgSim* and *MaxSim* similarity measurement of sense embeddings and synset embeddings are listed in Table 2. For word embeddings, each word pair has one word similarity since each word has one word embedding. The Spearman correlation $\rho$ of word embeddings is 0.471, which is not listed in Table 2.

**Table 2.** Spearman correlation $\rho$ based on Tongyici Cilin.

| Embeddings | *AvgSim* | *MaxSim* |
|---|---|---|
| Sense embeddings | **0.479** | **0.517** |
| Synset embeddings | 0.161 | 0.161 |

From Table 2, we can see that:

- Sense embeddings significantly outperform word embeddings and synset embeddings using either *MaxSim* or *AvgSim* measurement. Sense embeddings could capture the differences senses of polysemous words and therefore get the best performance.
- Synset embeddings get the worst results. The reason is that synset embeddings of the word pair appears in the same synset are the same, which fails to distinguish synonyms.

The similarity scores of sense embeddings using *MaxSim* measurement are used as the lexicon-based similarity scores for the combination of the two methods in the following experiments.

**The Effects of the Combination Strategies.** Table 3 shows the similarity results of the 5 combination strategies. From Table 3 we can see that all of the combination strategies could improve the performance significantly. Average strategy achieves the best $\rho$ values of 0.578 among all of the combination strategies. This is because it could balance between the similarity results of the two methods which belong to numerical data. This indicates that the lexicon-based and corpus-based methods are both effective and complementary for capturing the senses of polysemous words.

**Table 3.** Combination results based on 5 strategies.

| No. | Strategy | $\rho$ |
|---|---|---|
| 1 | Max | 0.561 |
| 2 | Average | **0.578** |
| 3 | Max and average | 0.556 |
| 4 | Replace 0 | 0.576 |
| 5 | Improved geometric mean | 0.565 |

We further investigate the impact of the parameters $\alpha$ and $\beta$ that control the weighting of synset constraints vs. sense constraints vs. word relation constraints. The Average and the Replace 0 strategies are employed as these two strategies show better results than other strategies.

As shown in Fig. 4, the best performance weighting area of the 2 combination strategies are both in the center and therefore all the three constraints are effective for sense embedding learning. It should be noted that average strategy achieves the best $\rho$ values of 0.582 with the parameters $\alpha = 0.5$ and $\beta = 0.4$. This indicates that the lexicon-based and corpus-based methods are effective and complementary for capturing the multiple senses of polysemous words.

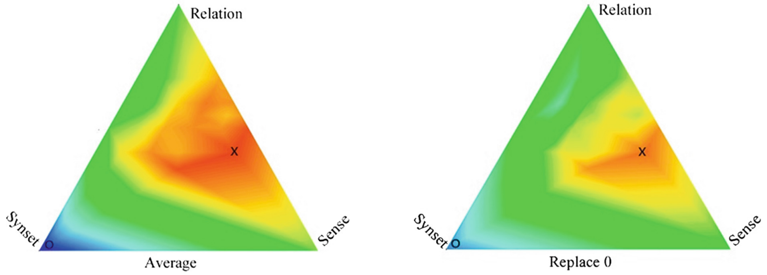

**Fig. 4.** Performance of different combination strategies with adjusting weightings of three constraints. “x” indicates the maximum and “o” indicates the minimum.

### 4.3 Compare with Related Work

As can be seen in Table 4, compared with word embedding methods, our method achieves a significant improvement on the baseline system of GloVe [13] and other methods such as Skip-Gram model of Word2vec [11, 12]. This indicates that sense embeddings is more beneficial to polysemous words similarity computation than single word embeddings. For other sense embedding methods, Huang et al. [4] cluster the contexts of polysemous word with a uniform parameter $K$ (the number of clusters), and then learn embeddings of the sense-labeled words. Comparing with Huang et al. [4], our corpus-based method clusters the effective context representations with the real number of polysemous senses, and achieves a $\rho$ value of 0.522. Guo et al. [9] propose a novel approach for Chinese sense embeddings leaning by exploiting bilingual parallel data. Their method heavily depends on high quality bilingual parallel data. Benefiting from the complementary relationship of corpus-based and lexicon-based methods, we improve the $\rho$ value to 0.582 with Average combination strategy.

**Table 4.** Spearman correlation $\rho$ on the dataset compared with the previous methods.

| Methods | $\rho$ |
|---|---|
| GloVe (*baseline*) | 0.492 |
| Word2vec | 0.497 |
| Huang et al. [4] | 0.407 |
| Guo et al. [9] | 0.554 |
| Corpus-based | 0.522 |
| Lexicon-based | 0.517 |
| Combination | **0.582** |

## 5 Conclusions

This paper proposes a framework for Chinese polysemous words similarity measurement, including a corpus-based method, a lexicon-based method. Evaluation on the Chinese Polysemous Word Similarity Dataset shows both the two methods could capture the senses of polysemous words well. Furthermore, we investigate the complement relation of the two methods and improve the $\rho$ value up to 0.582 by combining the two methods, which outperforms the state-of-the-art performance on the dataset. In the future, we would like to exploit more rich knowledge resources for sense embedding learning and investigate more effective combining strategies for Chinese polysemous words similarity computation.

**Acknowledgements.** This research is supported by Natural Science Foundation of China (No. 61272375).

## References

1. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of ACL, pp. 384–394 (2010)
2. Li, J., Jurafsky, D.: Do multi-sense embeddings improve natural language understanding. In: Proceedings of EMNLP, pp. 1722–1732 (2015)
3. Reisinger, J., Mooney, R.J.: Multi-prototype vector-space models of word meaning. In: Proceedings of NAACL-HLT, pp. 109–117 (2010)
4. Huang, E.H., Socher, R., Manning, C.D., Ng, A.Y.: Improving word representations via global context and multiple word prototypes. In: Proceedings of ACL, pp. 873–882 (2012)
5. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**(5), 513–523 (1988)
6. Dong, Z.D., Dong, Q.: HowNet and the computation of meaning. In: World Scientific, pp. 85–95 (2006)
7. Rothe, S., Schütze, H.: Autoextend: extending word embeddings to embeddings for synsets and lexemes. In: Proceedings of ACL, pp. 1793–1803 (2015)
8. Che, W.X., Li, Z.H., Liu, T.: LTP: a Chinese language technology platform. In: Proceedings of COLING, pp. 13–16 (2010)
9. Guo, J., Che, W.X., Wang, H.F., Liu, T.: Learning sense-specific word embeddings by exploiting bilingual resources. In: Proceedings of COLING, pp. 497–507 (2014)
10. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Workshop at ICLR (2013)
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111–3119 (2013)
13. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of EMNLP, pp. 1532–1543 (2014)
14. Neelakantan, A., Shankar, J., Passos, A., McCallum, A.: Efficient non-parametric estimation of multiple embeddings per word in vector space. In: Proceedings of EMNLP, pp. 1059–1069 (2014)

15. Zheng, X.Q., Feng, J.T., Chen, Y., Peng, H.Y., Zhang, W.Q.: Learning context-specific word/character embeddings. In: Proceedings of the AAAI 2017, pp. 3393–3399 (2017)
16. Chen, T., Xu, R.F., He, Y.L., Wang, X.: Improving distributed representation of word sense via WordNet gloss composition and context clustering. In: Proceedings of ACL, pp. 15–20 (2015)
17. Pei, J.H., Zhang, C., Huang, D.G., Ma, J.J.: Combining word embedding and semantic lexicon for Chinese word similarity computation. In: Proceedings of NLPCC, pp. 766–777 (2016)